What is claimed is:

1. A method for processing a database command, performed by an alternate database engine, the method comprising:

receiving, from a user, a database command requiring data from a first database engine, the first database engine having a command layer for processing database commands; and

processing the database command using only a command layer of an alternate database engine without accessing the command layer of the first database engine.

2. The method of claim 1, wherein the first database engine stores the data in a first database file

3. The method of claim 1, wherein the alternate database engine stores second data in a second database file.

4. The method of claim 1, wherein the database command is compatibile with at least one of: a Structured Query Language format, a Javascript Database Connectivity (JDBC) protocol and an Open-Database Connectivity protocol.

5. The method of claim 1, wherein the database command is a query.

6. The method of claim 5, said processing the database command further comprising:

evaluating the query.

7. The method of claim 6, said evaluating further comprising:

evaluating the query against system usage prior to submission to the alternate database engine.

8. The method of claim 7, said evaluating further comprising:

evaluating the query based on at least one of: a parameter of the query, a number of relational databases to be accessed for the query, a size of a data field to be searched for the query, an availability of resources of a system maintaining the alternate database engine, an availability of resources of a system maintaining the first database engine, a number of relational database tables to be employed for the query, a limitation imposed on a size of a query result set, a number of columns of data to be returned in a query result set, a cost of a similar stored query and a number of function calls for the query.

9. The method of claim 7, further comprising:

submitting the query to the alternate database engine with a limit on a number of returns responsive to the query, based on said evaluating.

10. The method of claim 7, further comprising

editing the query, based on said evaluating.

11. The method of claim 7, further comprising:

rejecting the query, based on said evaluating.

12. The method of claim 6, wherein said evaluating comprising:

determining, prior to said processing, whether the database command requires accessing the first database engine, and if not, accessing data stored only by the alternate database engine.

13. The method of claim 12, said determining further comprising:

translating the query to a native format of the alternate database engine.

14. The method of claim 6, said evaluating further comprising:

determining whether the query requires accessing temporally sensitive data, and if so, accessing a transaction log of the first database engine.

15. The method of claim 5, further comprising:

generating a result of the query

16. The method of claim 15, further comprising:

transmitting the result to the one of the plurality of users submitting the database command.

17. The method of claim 16, wherein said transmitting further comprises:

transmitting the result in a format of the first database engine.

18. The method of claim 1, further comprising:

storing second data in a database file maintained by the alternate database engine.

19. The method of claim 18, said processing further comprising:

determining whether the database command requires at least a portion of said second data, and if so, identifying said portion responsive to the database command.

20. The method of claim 1, further comprising:

receiving new data to be provided responsive to database commands; and

storing said new data in a database file maintained by the alternate database engine.

21. The method of claim 1, further comprising:

receiving new data to be provided responsive to database commands; and

storing said new data in a database file maintained by the first database engine.

22. The method of claim 1, said processing further comprising:

translating the database command to a native format of the alternate database engine.

23. The method of claim 1, wherein said processing further comprises:

identifying data stored by the first database engine that is responsive to the database command; and

accessing said identified data, wherein said identifying and accessing are performed exclusively through the command layer of the alternate database engine, without interaction with the command layer of the first database engine.

24. The method of claim 1, wherein the alternate database engine executes only read-only databases commands.

25. An apparatus for processing a database command, comprising:

means for receiving, from a user, a database command requiring data from a first database engine, the first database engine having a command layer for processing database commands; and

means for processing the database command using only a command layer of an alternate database engine without accessing the command layer of the first database engine.

26. An apparatus for processing a database command, comprising:

a processor; and

a memory in operative communication with the processor, the memory for storing a plurality of processing instructions for directing the processor to:

receive, from a user, a database command requiring data from a first database engine, the first database engine having a command layer for processing database

commands; and

process the database command using only a command layer of an alternate database engine without accessing the command layer of the first database engine.

27. A computer-readable medium encoded with processing instructions for implementing a method for processing a database command, performed by an alternate database engine, the method comprising:

receiving, from a user, a database command requiring data from a first database engine, the first database engine having a command layer for processing database commands; and

processing the database command using only a command layer of an alternate database engine without accessing the command layer of the first database engine.

28. A method for implementing and using an alternate database engine in conjunction with an established database engine, the method comprising:

providing access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands;

establishing an alternate database engine on the computing system;

receiving a database command from one of the plurality of users, the database command directed to data stored by the first database engine; and

processing the database command using only the alternate database engine without accessing the command layer of the first database engine.

29. The method of claim 28, wherein the computer system is at least one of: a local area network, a wide area network, an intranet, an extranet, a wireless network and the Internet.

30. The method of claim 28, wherein the first database engine stores the data in a first database file and the alternate database engine stores data in a second database file.

31. The method of claim 28, wherein the database command is compatible with at least one of: a Structured Query Language format, a Javascript Database Connectivity protocol and an Open-Database Connectivity protocol.

32. The method of claim 28, wherein the database command is a query.

33. The method of claim 32, said processing the database command further comprising:

evaluating the query.

34. The method of claim 33, said evaluating further comprising:

evaluating the query against system usage prior to submission to the alternative database engine.

35. The method of claim 34, said evaluating further comprising:

evaluating the query based on at least one of : a parameter of the query, a number of relational databases to be accessed for the query, a size of a data field to be searched for the

query, an availability of resources of a system maintaining the alternate database engine, an availability of resources of a system maintaining the first database engine, a number of relational database tables to be employed for the query, a limitation imposed on a size of a query result set, a number of columns of data to be returned in a query result set, a cost of a similar stored query and a number of function calls for the query.

36. The method of claim 34, further comprising:

submitting the query to the alternate database engine with a limit on a number of returns responsive to the query, based on said evaluating.

37. The method of claim 34, further comprising

editing the query, based on said evaluating.

38. The method of claim 34, further comprising:

rejecting the query, based on said evaluating.

39. The method of claim 33, wherein said evaluating comprising:

determining, prior to said processing, whether the database command requires accessing the data of the first database engine, and if not, accessing only data stored by the alternate database engine.

40. The method of claim 39, said determining further comprising:

translating the query to a native format of the alternate database engine.

41. The method of claim 33, said evaluating further comprising:

determining whether the query requires accessing temporally sensitive data, and if so, accessing a transaction log of the first database engine.

42. The method of claim 32, further comprising:

generating a result of the query

43. The method of claim 42, further comprising:

transmitting the result to the one of the plurality of users submitting the database command.

44. The method of claim 43, wherein said transmitting further comprises:

transmitting the result in a format of the first database engine.

45. The method of claim 28, further comprising further comprising:

storing second data in a database file maintained by the alternate database engine.

46. The method of claim 45, said processing further comprising:

determining whether the database command requires at least a portion of said second data, and if so, identifying said portion responsive to the database command.

47. The method of claim 28, further comprising:

receiving new data to be provided to the plurality of users; and

storing said new data in a database file maintained by the firsy database engine.

48. The method of claim 28, said processing further comprising:

translating the database command to a native format of the alternate database engine.

49. The method of claim 28, wherein said processing further comprises:

identifying data stored by the first database engine that is responsive to the database command; and

accessing said identified data, wherein said identifying and accessing are performed exclusively through a command layer of the alternate database engine, without interaction with the command layer of the first database engine.

50. The method of claim 28, wherein the alternate database engine executes only read-only databases commands.

51. An apparatus for implementing and using an alternate database engine in conjunction with an established database engine, comprising:

means for providing access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands;

means for establishing an alternate database engine on the computing system;

means for receiving a database command from one of the plurality of users, the database command directed to data stored by the first database engine; and

means for processing the database command using only the alternate database engine without accessing the command layer of the first database engine.

52. An apparatus for implementing and using an alternate database engine in conjunction with an established database engine, comprising:

a processor; and

a memory in operative communication with the processor, the memory for storing a plurality of processing instructions directing the processor to:

provide access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands;

establish an alternate database engine on the computing system;

receive a database command from one of the plurality of users, the database command directed to data stored by the first database engine; and

process the database command using only the alternate database engine without accessing the command layer of the first database engine.

53. A computer-readable medium encoded with processing instructions for performing a method of implementing and using an alternate database engine in conjunction with an established database engine, the method comprising:

providing access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands;

establishing an alternate database engine on the computing system;

receiving a database command from one of the plurality of users, the database command directed to data stored by the first database engine; and

processing the database command using only the alternate database engine without accessing the command layer of the first database engine.

54. A method for implementing and using an alternate database engine in conjunction with an established database engine, the method comprising:

providing access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands, the first database engine further maintaining a first database file including first data;

establishing an alternate database engine on the computing system, the alternate database engine maintaining a second database file including second data;

receiving a database command from one of the plurality of users, the database command directed to at least one of said first and second data; and

processing the database command using only the alternate database engine without accessing the command layer of the first database engine.

55. The method of claim 54, wherein the computer system is at least one of: a local area network, a wide area network, an intranet, an extranet, a wireless network and the Internet.

56. The method of claim 54, wherein the first database engine stores the data in a first database file and the alternate database engine stores data in a second database file.

57. The method of claim 54, wherein the database command is compatible with at least one of: a Structured Query Language format, a Javascript Database Connectivity protocol and an Open-Database Connectivity protocol.

58. The method of claim 54, wherein the database command is a query.

59. The method of claim 58, said processing the database command further comprising:

evaluating the query.

60. The method of claim 59, said evaluating further comprising:

evaluating the query against system usage prior to submission to the alternative database engine.

61. The method of claim 60, said evaluating further comprising:

evaluating the query based on at least one of : a parameter of the query, a number of relational databases to be accessed for the query, a size of a data field to be searched for the query, an availability of resources of a system maintaining the alternate database engine, an availability of resources of a system maintaining the first database engine, a number of relational database tables to be employed for the query, a limitation imposed on a size of a query result set, a number of columns of data to be returned in a query result set, a cost of a similar stored query and a number of function calls for the query.

62. The method of claim 60, further comprising:

submitting the query to the alternate database engine with a limit on a number of returns responsive to the query, based on said evaluating.

63. The method of claim 60, further comprising

editing the query, based on said evaluating.

64. The method of claim 60, further comprising:

rejecting the query, based on said evaluating.

65. The method of claim 59, wherein said evaluating comprising:

determining, prior to said processing, whether the database command requires accessing the first data, and if not, accessing only said second data using the alternate database engine.

66. The method of claim 65, said determining further comprising:

translating the query to a native format of the alternate database engine.

67. The method of claim 59, said evaluating further comprising:

determining whether the query requires accessing the temporally sensitive data of said first data, and if so, accessing a transaction log of the first database engine.

68. The method of claim 58, further comprising:

generating a result of the query

69. The method of claim 68, further comprising:

transmitting the result to the one of the plurality of users submitting the database command.

70. The method of claim 69, wherein said transmitting further comprises:

transmitting the result in a format of the first database engine.

71. The method of claim 54, further comprising:

receiving new data to be provided to the plurality of users; and

storing said new data in the first database file.

72. The method of claim 54, further comprising:

receiving new data to be provided to the plurality of users; and

storing said new data in the second database file.

73. The method of claim 54, said processing further comprising:

translating the database command to a native format of the alternate database engine.

74. The method of claim 54, wherein said processing further comprises:

identifying data stored by the first database engine that is responsive to the database command; and

accessing said identified data, wherein said identifying and accessing are performed exclusively through a command layer of the alternate database engine, without interaction with the command layer of the first database engine.

75. The method of claim 54, wherein the alternate database engine executes only read-only databases commands.

76. An apparatus for implementing and using an alternate database engine in conjunction with an established database engine, comprising:

means for providing access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands, the first database engine further maintaining a first database file including first data;

means for establishing an alternate database engine on the computing system, the alternate database engine maintaining a second database file including second data;

means for receiving a database command from one of the plurality of users, the database command directed to at least one of said first and second data; and

means for processing the database command using only the alternate database engine without accessing the command layer of the first database engine.

77. An apparatus for implementing and using an alternate database engine in conjunction with an established database engine, comprising:

a processor; and

a memory in operative communication with the processor, the memory for storing a plurality of processing instructions directing the processor to:

provide access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands, the first database engine further maintaining a first database file including first data;

establish an alternate database engine on the computing system, the alternate database engine maintaining a second database file including second data;

receive a database command from one of the plurality of users, the database command directed to at least one of said first and second data; and

process the database command using only the alternate database engine without accessing the command layer of the first database engine.

78. A computer-readable medium encoded with processing instructions for performing a method of implementing and using an alternate database engine in conjunction with an established database engine, the method comprising:

providing access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands, the first database engine further maintaining a first database file including first data;

establishing an alternate database engine on the computing system, the alternate database engine maintaining a second database file including second data;

receiving a database command from one of the plurality of users, the database command directed to at least one of said first and second data; and

processing the database command using only the alternate database engine without accessing the command layer of the first database engine.

79. A method for requesting data, performed by a user on a computing system, the method comprising:

transmitting a database command directed to data stored by a first database engine, the first database engine having a command layer for processing database commands; and

receiving a result of the database command from an alternate database engine, wherein the database command is processed by the alternate database engine without accessing the command layer of the first database engine.

80. A method for requesting data, performed by a user on a computing system, the method comprising:

transmitting a database command directed to data stored by a first database engine and an alternate database engine, the first database engine having a command layer for processing database commands; and

receiving a result of the database command from an alternate database engine, wherein the database command is processed by the alternate database engine without accessing the command layer of the first database engine.

81. A method for processing a database command, performed by an alternate read-only database engine, the method comprising:

receiving, from a user, a read-only database command requiring data from a first database engine, the first database engine having a command layer for processing database commands; and

processing the database command using only a command layer of an alternate database engine without accessing the command layer of the first database engine and without executing any write commands and read-write commands.

82. A method for implementing and using an alternate database engine in conjunction with an established database engine, the method comprising:

providing access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands;

establishing an alternate database engine on the computing system;

receiving a read-only database command from one of the plurality of users, the read-only database command directed to data stored by the first database engine; and

processing the database command using only the alternate database engine without accessing the command layer of the first database engine, wherein the alternate database engine executes read-only database commands.

83. A method for implementing and using an alternate database engine in conjunction with an established database engine, the method comprising:

providing access to a first database engine to a plurality of users on a computing system, the first database engine having a command layer for processing database commands, the first database engine further maintaining a first database file including first data;

establishing an alternate database engine on the computing system, the alternate database engine maintaining a second database file including second data;

receiving a read-only database command from one of the plurality of users, the read-only database command directed to at least one of said first and second data; and

processing the database command using only the alternate database engine without accessing the command layer of the first database engine, wherein the alternate database engine executes only read-write database commands.

84. A method for processing a database command, performed by an alternate database engine, the method comprising:

receiving, from a user, a database command directed to a first database engine, the first database engine having a command layer for processing database commands;

processing the database command using only a command layer of the alternative database engine without accessing the command layer of the first database engine, said processing further comprising:

evaluating the database command to determine system usage of the query at the database engine, prior to execution of the database command, said evaluating based on at least one of : a parameter of the , a number of relational databases for the database command, a size of a data field to be searched for the database command, an availability of resources of the

database engine, a number of relational database tables to be employed for the database command, a limitation imposed on a size of a query result set, a number of columns of data to be returned in a query result set, a cost of a similar stored database command and a number of function calls for the database command;

determining a threshold value for system usage of the alternate database engine, wherein the threshold value is based on at least one of: estimated processor usage, estimated memory usage, input/output resource usage and disk resource usage of the alternate database engine;

if the system usage surpasses a threshold value, performing at least one of the following: submitting the database command to the alternate database engine with a limit on a number of returns responsive to the database command, editing the database command, and rejecting the database command;

determining whether the database command requires accessing data maintained by the first database engine, and if not, accessing second data stored only by the alternate database engine;

determining whether the database command requires accessing temporally sensitive data, and if so, accessing a transaction log of the first database engine;

translating the database command to a native format of the alternate database engine;

generating a result of the database command; and

transmitting the result to the user in a format of the first database engine.